

Constraint satisfaction problems with isolated solutions are hard

Lenka Zdeborová

*Université Paris-Sud, LPTMS, UMR8626, Bât. 100, Université Paris-Sud 91405 Orsay cedex
CNRS, LPTMS, UMR8626, Bât. 100, Université Paris-Sud 91405 Orsay cedex and
Theoretical Division and Center for Nonlinear Studies,
Los Alamos National Laboratory, Los Alamos, NM 87545*

Marc Mézard

*Université Paris-Sud, LPTMS, UMR8626, Bât. 100, Université Paris-Sud 91405 Orsay cedex and
CNRS, LPTMS, UMR8626, Bât. 100, Université Paris-Sud 91405 Orsay cedex*

(Dated: December 4, 2008)

We study the phase diagram and the algorithmic hardness of the random ‘locked’ constraint satisfaction problems, and compare them to the commonly studied ‘non-locked’ problems like satisfiability of boolean formulas or graph coloring. The special property of the locked problems is that clusters of solutions are *isolated* points. This simplifies significantly the determination of the phase diagram, which makes the locked problems particularly appealing from the mathematical point of view. On the other hand we show empirically that the clustered phase of these problems is extremely hard from the algorithmic point of view: the best known algorithms all fail to find solutions. Our results suggest that the easy/hard transition (for currently known algorithms) in the locked problems coincides with the clustering transition. These should thus be regarded as new benchmarks of really hard constraint satisfaction problems.

PACS numbers: 89.70.Eg,75.10.Nr,64.70.P-

I. INTRODUCTION

Constraint satisfaction problems (CSPs) play a crucial role in theoretical and applied computer science. Their wide range of applicability arises from their very general nature: given a set of N discrete variables subject to M constraints, a CSP consists in deciding whether there exists an assignment of variables which satisfies simultaneously all the constraints. When such an assignment exists we call it a solution and aim at finding it. One of the most important questions about a CSP is how hard it is to find a solution or prove that there is none. Many of the CSPs belong to the class of NP-complete problems [1, 2]. This basically means that, if $P \neq NP$, there is no algorithm able to solve the worst case instances of the problem in a polynomial time. Next to the question of the *worst case* computational complexity arises the less explored question of *typical case* complexity. A pivotal step in understanding the typical case complexity is the study of *random* CSPs where each constraint involves a finite number of variables. Pioneering work on this subject [3, 4] discovered that many problems are empirically harder close to the so-called satisfiability phase transition. This is a phase transition appearing at a critical constraint density α_s such that for $M/N = \alpha < \alpha_s$ almost every large instance of the problem has at least one solutions, and for $\alpha > \alpha_s$ almost all large instances have no solution.

Studies of phase transitions such as the one occurring in the satisfiability problem are natural for statistical physicists. Indeed the methods developed to study frustrated disordered systems like glasses and spin glasses [5] have turned out to be very fruitful in the study of several CSPs. In particular they allow some structural studies which aim at understanding how the difficulty of a problem is related to the geometrical organization of its solutions. Several other phase transitions were described in this context. The most important one is probably the clustering transition [6, 7], known as the dynamical glass transition in the mean field theory of glasses. It was computed that in the region where the density of constraints is below the satisfiability threshold there exists a phase where the space of solutions splits into ergodically separated groups – clusters. Another important property of the clusters concerns the freezing of the variables. A variable is frozen in a cluster if it takes the same value in all the solutions of this cluster. It has been conjectured that the clustering [8] and the freezing of variables [9] are two ingredients which contribute to make a random CSP hard. But the predictions for the easy/hard transition in a general random CSP are still not fully quantitative. The present work provides further insight into this subject.

In this paper we present a detailed study of the *locked* CSPs, introduced recently in [10]. The special property of the locked problems is that clusters are point-like: every cluster contains only one solution. Therefore, as soon as the system is in a clustered phase, all the variables are frozen in each cluster. The clustering and the freezing phase transitions occur simultaneously. Consequently the organization of the space of solutions is much simpler than in the commonly studied K-satisfiability or graph coloring [6, 9, 11, 12, 13]. But at the same time, and unlike in the K-satisfiability or graph coloring problems, the whole clustered phase is extremely hard for all existing algorithm and

the clustering/freezing threshold seems to coincide very precisely with the onset of this hardness.

The interest in the locked problems is thus twofold:

- (a) **Locked problems are very simple:** As the clusters of solutions are point-like many of the quantities of interest can be computed using simpler tools than in the canonical K -satisfiability problem. This is in particular interesting from the mathematical point of view, because several of their properties become accessible to rigorous proofs. From a broader point of view the locked problems should be useful as simple models of glass forming liquids because their phase diagram can be studied without any need to introduce the complicated scheme of ‘replica symmetry breaking’ [5].
- (b) **Locked problems are very hard:** From the algorithmic point of view the whole clustered phase of the locked problems is extremely hard, none of the known algorithms is able to find solutions efficiently. This suggests to use locked CSPs as hard benchmarks. At the same time one may hope that the performance of some algorithms will be simpler to analyze when they are applied to the locked problems, compared to the general case.

This paper is organized as follows: In section II we define the random occupation problems and the random locked occupation problems (LOPs) on which we will illustrate our main findings. In section III we write the equations needed to describe the phase diagram of the occupation problems, using well known tools from statistical physics and probability theory. In section IV we summarize the basic properties of the phase diagram in general random CSPs and then discuss in detail the situation in the locked problems. We also discuss the class of so-called balanced LOPs which are even simpler from the mathematical point of view. Finally section V shows our findings about algorithmic performance in the occupation problems: empirical data using the best known random CSP solver – belief propagation reinforcement – indicates that the clustering threshold is close to the boundary between the easy and hard regions. We analyze also the non-locked occupation problems for comparison. A short summary of the results and perspectives conclude the paper in section VI.

II. DEFINITIONS

A. Locked occupation problems

We shall study a broad class of problems called ‘*occupation problems*’. An occupation problem involves N binary variables $s_i \in \{0, 1\}$ ($s_i = 0$ is referred to as “site i is empty”, and $s_i = 1$ is “occupied”) and M constraints, indexed by $b \in \{1, \dots, M\}$. Each constraint b involves K_b distinct variables, and is defined by a ‘constraint word’ A^b with $K_b + 1$ bits, which we write as $A^b = A_0^b A_1^b \dots A_{K_b}^b$, where $A_i^b \in \{0, 1\}$. We denote by ∂b the indices of all variables involved in the constraint b . The constraint b is satisfied if and only if the sum $r = \sum_{i \in \partial b} s_i$ of all its variables is such that $A_r^b = 1$. In other words, in order for constraint b to be satisfied, one needs that the number of occupied sites, r , in its neighborhood, must be such that $A_r^b = 1$ (this unified notation for the occupation problems was introduced in [14]).

Definition: An occupation problem is *locked* if and only if:

- (a) For every constraint $b \in \{1, \dots, M\}$, the vector A^b is such that, for all $i = 0, \dots, K - 1$: $A_i^b A_{i+1}^b = 0$.
- (b) Every variable appears in at least two different constraints.

In this paper, we shall study only ‘constraint-regular’ problems in which all of the constraints are described by the same constraint word: for all $b \in \{1, \dots, M\}$, $K_b = K$ and $A^b = A$. Furthermore, in order to focus onto difficult cases, we shall only consider the occupation problems where neither the totally empty nor the totally occupied configurations are solution, i.e. we keep to the cases where $A_0 = A_K = 0$. It is convenient to use the factor graph description of a problem [15, 16], where sites and constraints are vertices, and an edge connects a constraint a to a site i whenever i appears in constraint a (see Fig. 1). An instance of a constraint-regular occupation model is fully described by its factor graph (where all constraint vertices have degree K) and the $K + 1$ component vector A . The locked problems are thus characterized by the facts that (i) there are no consecutive ‘1’ in the word $A = A_0 A_1 \dots A_K$, and (ii) their factor graph has no leaves.

Well-studied examples of occupation problems include:

- Ising anti-ferromagnet: $A = 010$
- Odd parity checks (anti-ferromagnetic K -spin model, with K even): $A = 01010 \dots 1010$
- Positive 1-in- K SAT (exact cover): $A = 0100 \dots 00$ [17]

- Perfect matching in K -regular graphs: each variable belongs to two constraints and $A = 01000\dots 00$ [18]
- Bicoloring (positive NAE-SAT): $A = 0111\dots 110$ [19, 20, 21]
- Circuits going through all the points: $A = 001000\dots 00$ [22]

All these examples, except the bicoloring, are locked on graphs without leaves.

For the occupation problems which have not been studied previously, we will use names derived in the following way: $A = 010100$ is the 1-or-3-in-5 SAT, $A = 010010$ is the 1-or-4-in-5 SAT, etc.

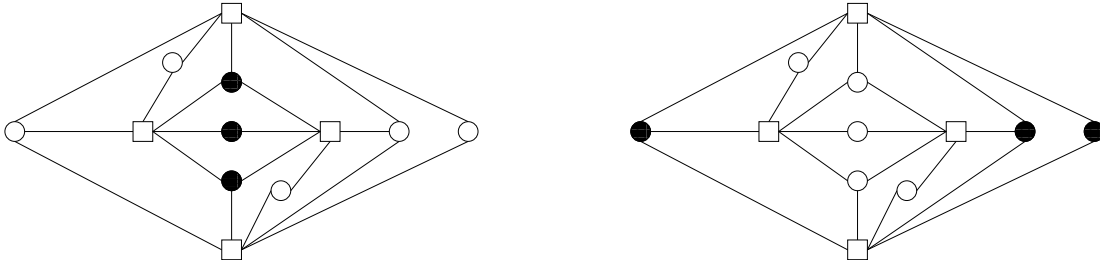


FIG. 1: A factor graph representation of an instance of the 1-or-3-in-5 SAT ($A = 010100$). The squares are the constraints. Full/empty circles are occupied/empty sites. The two parts show two examples of satisfying assignments – “solutions” – of this instance. As there are no leaves (each variable belongs to at least two constraints) and A satisfies $A_i A_{i+1} = 0$ for all $i = 0, \dots, 4$, this instance is locked.

From the computational complexity point of view, Schaefer’s theorem [23] implies that most of the occupation problems are NP-complete. The exceptions are the parity checks, which amount to linear systems of equations on $GF(2)$, and some of the cases where the variables have degree 2, such as for instance the perfect matching.

The crucial property of the locked occupation problems is that in order to go from one solution to another one must flip at least a closed loop of variables. This property can be used to generalize the definition of a locked problems to a much wider class of constraint satisfaction problems than the occupation problems, and in particular the variables do not need to be binary. Some examples of locked problems which are not occupation problems are the XOR-SAT (p-spin) problem on factor graphs without leaves [24], or all the uniquely extensible models [25].

B. Ensembles of random occupation problems

We shall study some random ensembles of locked occupation problems, in which the factor graph is chosen from some ensemble of random bipartite graphs. We consider constraint-regular occupation problems where each constraint involves K variables, and is characterized by the constraint word A . An ensemble is characterized via a probability distribution $Q(l)$. To create an instance of the random occupation problem with N variables, we draw N independent random numbers l_i from the distribution $Q(l)$, with the additional constraint that $\sum_{i=1}^N l_i/K = M$ is an integer. The factor graph that characterizes an instance is then chosen uniformly at random from all the possible graphs with N variables, and M constraints, such that, for all $i = 1 \dots N$, the variable i is connected to l_i constraints.

In this paper we will consider mainly two degree distributions:

- Regular: $Q(l) = \delta_{l,L}$, in which all the variables take part in L clauses.
- Truncated Poissonian:

$$Q(0) = Q(1) = 0, \quad Q(l) = \frac{1}{1 - (1+c)e^{-c}} \frac{e^{-c} c^l}{l!} \quad \text{for } l \geq 2 \quad (1)$$

where $c \geq 0$. The average “connectivity” (variable degree) is then

$$\bar{l} = c \frac{1 - e^{-c}}{1 - (1+c)e^{-c}}. \quad (2)$$

In the cavity method one also needs the excess degree distribution $q(l)$, defined as the distribution of the number of neighbors on one side of an edge chosen uniformly at random:

$$q(0) = 0, \quad q(l) = \frac{1}{e^c - 1} \frac{c^l}{l!}. \quad (3)$$

We shall be interested in the properties of large instance, i.e. in the ‘thermodynamic limit’ where one sends $N \rightarrow \infty$ and $M \rightarrow \infty$, keeping K and $Q(l)$ fixed; this results in a fixed density of constraints $M/N = \bar{l}/K$. Our main results are easily generalizable to any degree distribution $Q(l)$ which has a finite second moment. For every such distribution, a typical factor graph is locally tree-like: the shortest loop going through a typical variable has a length which scales as $\log N$. The crucial property of the locked occupation problems is that, in order to go from one solution of the problem to another solution, one must flip at least one closed loop of variables. On the random locally tree-like factor graphs this means that at least $\log N$ variables need to be changed.

III. THE SOLUTION OF RANDOM OCCUPATION PROBLEMS

The cavity method [26] is nowadays the standard tool to compute the phase diagram of random locally tree-like constraint satisfaction problems. Depending on the structure of the space of solutions of the problem, different versions (levels of the replica symmetry breaking) of the method are needed. In this section we state the cavity equations for the occupation problems. For a detailed derivation and discussion of the method see [16, 26].

We index the variables by i, j, k, \dots going from 1 to N , and the constraints by a, b, c, \dots going from 1 to M . The energy of the occupation problems then reads

$$H(\{s\}) = \sum_{a=1}^M \delta_{A_{\sum_j \in \partial a} s_j, 0}, \quad (4)$$

In this paper we shall study only the instances where solutions (ground states of zero energy) exist, and we shall focus on the uniform measure over all solutions.

A. The replica symmetric solution

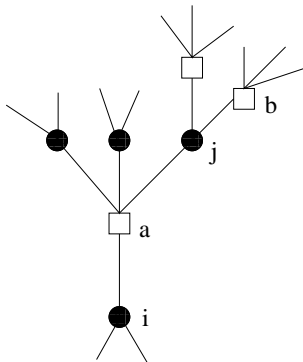


FIG. 2: Part of the factor graph to illustrate the meaning of indices in the belief propagation equations (5a-5b).

The replica symmetric version of the cavity method is also known under the name belief propagation [15, 16, 27]. It exploits the local tree-like property of the factor graph, assuming that correlations decay fast enough. The basic quantities used in this approach are messages. We define $\psi_{s_i}^{a \rightarrow i}$ as the probability that the constraint a is satisfied, conditioned to the fact that the value of variable i is s_i . Similarly, $\chi_{s_j}^{j \rightarrow a}$ is the probability that the variable j takes value s_j conditioned to the fact that the constraint a has been removed from the graph. The messages then satisfy the belief propagation (BP) equations

$$\psi_{s_i}^{a \rightarrow i} = \frac{1}{Z^{a \rightarrow i}} \sum_{\{s_j\}} \delta_{A_{s_i + \sum_j s_j, 1}} \prod_{j \in \partial a - i} \chi_{s_j}^{j \rightarrow a}, \quad (5a)$$

$$\chi_{s_j}^{j \rightarrow a} = \frac{1}{Z^{j \rightarrow a}} \prod_{b \in \partial j - a} \psi_{s_j}^{b \rightarrow j}, \quad (5b)$$

where $Z^{a \rightarrow i}$ and $Z^{j \rightarrow a}$ are normalization constants. Fig. 2 shows the corresponding part of the factor graph. The

marginal probabilities (“beliefs”) are then expressed as

$$\chi_{s_i}^i = \frac{1}{Z^i} \prod_{a \in \partial i} \psi_{s_i}^{a \rightarrow i}, \quad (6)$$

The replica symmetric entropy (logarithm of the number of solutions, divided by the system size) then reads

$$s = \frac{1}{N} \sum_a \log(Z^a) - \frac{1}{N} \sum_i (l_i - 1) \log(Z^i). \quad (7)$$

where

$$Z^a = \sum_{\{s_i\}} \delta_{A_{\Sigma_i s_i}, 1} \prod_{i \in \partial a} \left(\prod_{b \in \partial i - a} \psi_{s_i}^{b \rightarrow i} \right), \quad (8a)$$

$$Z^i = \prod_{a \in \partial i} \psi_0^{a \rightarrow i} + \prod_{a \in \partial i} \psi_1^{a \rightarrow i}, \quad (8b)$$

are the exponentials of the entropy shifts when the node a and its neighbors (resp. the node i) is added.

When one considers an ensemble of random graphs, the probability distribution of the messages can be found via the population dynamics technique [26]. Moreover, on the regular graph ensemble or for some of the balanced problems (see Sec. IV C) the solution is *factorized*. In the factorized solution the messages $\chi^{i \rightarrow a}$, $\psi^{a \rightarrow i}$ are independent of the edge (ia) and the replica symmetric solution can thus be found analytically.

For instance in the regular graph ensemble where each variable is present in L constraints the factorized solution is

$$\psi_0 = \frac{1}{Z^{\text{reg}}} \sum_{r=0}^{K-1} \delta_{A_r, 1} \binom{K-1}{r} \psi_1^{(L-1)r} \psi_0^{(L-1)(K-1-r)}, \quad (9a)$$

$$\psi_1 = \frac{1}{Z^{\text{reg}}} \sum_{r=0}^{K-1} \delta_{A_{r+1}, 1} \binom{K-1}{r} \psi_1^{(L-1)r} \psi_0^{(L-1)(K-1-r)}, \quad (9b)$$

where the normalization Z^{reg} is fixed by the condition $\psi_0 + \psi_1 = 1$. Given the solution ψ_0, ψ_1 of (9a-9b), the entropy reads

$$s_{\text{reg}} = \frac{L}{K} \log \left[\sum_{r=0}^K \delta_{A_r, 1} \binom{K}{r} \psi_1^{(L-1)r} \psi_0^{(L-1)(K-r)} \right] - (L-1) \log [\psi_0^L + \psi_1^L]. \quad (10)$$

B. Reconstruction on trees

Treating the locally tree-like random graph as a tree fails if long range correlations are present in the system. More precisely [12, 28] the replica symmetric assumption is correct if and only if the so-called point-to-set correlations do decay to zero. The decay of these correlations is closely related [28] to the problem of reconstruction on trees [29] which we explain and analyze in this section.

The reconstruction on trees is defined as follows: First construct a tree of d generations having the same connectivity properties as a finite neighborhood of a random variable in the random factor graph. Assign the root a random value, further assign values iteratively on the descendants uniformly at random but in such a way that the constraints are satisfied. Subsequently forget the assignment everywhere but on the leaves of the tree. The reconstruction on the tree is possible if and only if the information left in the values of the leaves about the value of the root does not go to zero as the size of the tree grows, $d \rightarrow \infty$. The replica symmetric assumption is correct if and only if the reconstruction is not possible, in other words if there is no correlation between the root (point) and the leaves (set). Typically, when the average connectivity of variables is small the reconstruction is not possible and when the connectivity is large the reconstruction is possible. The threshold connectivity is called the reconstructibility threshold, or the clustering transition. The clustering is then defined as a minimal decomposition of the space of solutions such that within the components (clusters) the point-to-set correlations do decay to zero [12].

It was shown in [28] that the analysis of the reconstruction on trees is equivalent to the solution of the one-step replica symmetry breaking (1RSB) equations at the value of the Parisi parameter $m = 1$ [26].

Instead of the general form of the 1RSB equations at $m = 1$ (see e.g. [30]), we shall only discuss here a simpler form called the *naive reconstruction* in [31]. In general the naive reconstruction gives only an upper bound on the reconstructibility threshold, but in the locked problems it gives in fact the full information. The naive reconstruction consists in computing the probability that the value of the root is *uniquely* implied by the leaves (boundary conditions). Here we give the equations only for regular graph ensembles with variables of connectivity L , where the factorized replica symmetric solution (9) holds. Define μ_1 (resp. μ_0) as the probability that a variable which in the broadcasting had value 1 (resp. 0) is uniquely determined by the boundary conditions. One has:

$$\mu_1 = \frac{1}{\psi_1 Z^{\text{reg}}} \sum_{r=0}^k \delta_{A_{r+1},1} \delta_{A_r,0} g_k(r) \sum_{s=0}^{s_1} \binom{r}{s} [1 - (1 - \mu_0)^l]^{k-r} [1 - (1 - \mu_1)^l]^{r-s} (1 - \mu_1)^{ls}, \quad (11a)$$

$$\mu_0 = \frac{1}{\psi_0 Z^{\text{reg}}} \sum_{r=0}^k \delta_{A_{r+1},0} \delta_{A_r,1} g_k(r) \sum_{s=0}^{s_0} \binom{k-r}{s} [1 - (1 - \mu_1)^l]^r [1 - (1 - \mu_0)^l]^{k-r-s} (1 - \mu_0)^{ls}, \quad (11b)$$

where $l = L - 1$, $k = K - 1$, and $g_k(r) = \binom{k}{r} (\psi_1)^{lr} (\psi_0)^{l(k-r)}$. The indices s_1, s_0 in the second sum of both equations are the largest possible but such that $s_1 \leq r$, $s_0 \leq K - 1 - r$, and $\sum_{s=0}^{s_1} A_{r-s} = 0$, $\sum_{s=0}^{s_0} A_{r+1+s} = 0$. The values ψ_0, ψ_1 are the fixed point of eqs. (9a-9b), and Z^{reg} is the corresponding normalization.

These lengthy equations have in fact a simple meaning. The first sum is over the possible numbers r of occupied variables on the descendants in the broadcasting. The sum over s is over the number of variables which were not implied by at least one constraint but the configuration of implied variables nevertheless implies the outgoing value. The term $1 - (1 - \mu)^l$ is the probability that at least one constraint implies the variable, $(1 - \mu)^l$ is the probability that none of the constraints implies the variable.

C. Survey Propagation

Survey propagation is a special form of the 1RSB equations corresponding to the value of the Parisi parameter $m = 0$ [8]. The main assumption of the 1RSB approach is that the space of solutions splits into clusters (pure states). To each cluster corresponds one fixed point of BP equations. Survey propagation are then iterative equations for the following probabilities (surveys)

$$\text{Prob}(\chi_1^{i \rightarrow a} = 1, \chi_0^{i \rightarrow a} = 0) = p_1^{i \rightarrow a}, \quad \text{Prob}(\psi_1^{a \rightarrow i} = 1, \psi_0^{a \rightarrow i} = 0) = q_1^{a \rightarrow i}, \quad (12a)$$

$$\text{Prob}(\chi_1^{i \rightarrow a} = 0, \chi_0^{i \rightarrow a} = 1) = p_0^{i \rightarrow a}, \quad \text{Prob}(\psi_1^{a \rightarrow i} = 0, \psi_0^{a \rightarrow i} = 1) = q_0^{a \rightarrow i}, \quad (12b)$$

$$p_*^{i \rightarrow a} = 1 - p_1^{i \rightarrow a} - p_0^{i \rightarrow a}, \quad q_*^{a \rightarrow i} = 1 - q_1^{a \rightarrow i} - q_0^{a \rightarrow i}, \quad (12c)$$

where $q_{1/0}^{a \rightarrow i}$ is probability over clusters that clause a is satisfied only if variable i takes value 1/0, $q_*^{a \rightarrow i}$ is then the probability that clause a can be satisfied by both values 1 and 0, similarly $p_{1/0}^{i \rightarrow a}$ is probability that variable i have to take value 1/0 if the clause a is not present, $p_*^{i \rightarrow a}$ is probability that the variable i can take both values 1 and 0 when clause a is not present. The survey propagation equations are then written in two steps, first the update of p 's knowing q 's

$$p_1^{j \rightarrow a} = \frac{1}{\mathcal{N}^{j \rightarrow a}} \left[\prod_{b \in \partial_{j-a}} (q_1^{b \rightarrow j} + q_*^{b \rightarrow j}) - \prod_{b \in \partial_{j-a}} q_*^{b \rightarrow j} \right], \quad (13a)$$

$$p_0^{j \rightarrow a} = \frac{1}{\mathcal{N}^{j \rightarrow a}} \left[\prod_{b \in \partial_{j-a}} (q_0^{b \rightarrow j} + q_*^{b \rightarrow j}) - \prod_{b \in \partial_{j-a}} q_*^{b \rightarrow j} \right], \quad (13b)$$

$$p_*^{j \rightarrow a} = \frac{1}{\mathcal{N}^{j \rightarrow a}} \prod_{b \in \partial_{j-a}} q_*^{b \rightarrow j}, \quad (13c)$$

second the update of q 's knowing p 's

$$q_s^{a \rightarrow i} = \frac{1}{\mathcal{N}^{a \rightarrow i}} \left[\sum_{\{r_j\}} C_s(\{r_j\}) \prod_{j \in \partial_{a-i}} p_{r_j}^{j \rightarrow a} \right]. \quad (14a)$$

Here $\mathcal{N}^{j \rightarrow a}$ and $\mathcal{N}^{a \rightarrow i}$ are normalization constants, the indices s and r_j are in $\{1, 0, *\}$. The function C_1/C_0 (resp. C_*) takes values 1 if and only if the incoming set of $\{r_j\}$ forces the variable i to be occupied/empty (resp. let the variable i free), in all other cases the C 's are zero. More specifically, let us call n_1, n_0, n_* the number of indices 1, 0, * in the set $\{r_j\}$ then

- $C_1 = 1$ if and only if $A_{n_1+n_*+1} = 1$ and $A_{n_1+n} = 0$ for all $n = 0 \dots n_*$;
- $C_0 = 1$ if and only if $A_{n_1} = 1$ and $A_{n_1+1+n} = 0$ for all $n = 0 \dots n_*$;
- $C_* = 1$ if and only if there exists $m, n \in \{0 \dots n_*\}$ such that $A_{n_1+n} = A_{n_1+m+1} = 1$.

D. The first and the second moment

In this section we give the formulas for the first and second moment method in general occupation problems. This allows for a direct probabilistic study of the balanced locked occupation problems introduced below in Sec. IV C.

For a given instance (or factor graph), G , define as \mathcal{N}_G the number of solutions. The first moment is the average of \mathcal{N}_G over the graph ensemble, which can also be written as:

$$\langle \mathcal{N}_G \rangle = \sum_{\{\sigma\}} \text{Prob}(\{\sigma\} \text{ is SAT}) . \quad (15)$$

The ‘annealed entropy’ is then defined as $s_{\text{ann}} \equiv \log \langle \mathcal{N}_G \rangle / N$. It is an upper bound on the quenched entropy, $\langle \log \mathcal{N}_G \rangle / N$. In order to compute the first moment we divide variables into groups according to their connectivity and in each group we choose a fraction of occupied variables. The number of ways in which this can be done is then multiplied by the probability that such a configuration satisfies simultaneously all the constraints. After some algebra [30] we obtain the entropy of solutions with a fraction $0 \leq t \leq 1$ of occupied variables:

$$s_{\text{ann}}(t) = \sum_l Q(l) \log [1 + u(t)^l] + \frac{\bar{l}}{K} \log \left[\sum_{r=0}^K \delta_{A_r, 1} \binom{K}{r} \left(\frac{t}{u(t)} \right)^r (1-t)^{K-r} \right], \quad (16)$$

where $u(t)$ is the inverse of

$$t = \frac{1}{\bar{l}} \sum_l l Q(l) \frac{u^l}{1 + u^l}. \quad (17)$$

The annealed entropy is then $s_{\text{ann}} = \max_t s_{\text{ann}}(t)$.

The second moment of the number of solutions is defined as

$$\langle \mathcal{N}_G^2 \rangle = \sum_{\{\sigma_1\}, \{\sigma_2\}} \text{Prob}(\{\sigma_1\} \text{ and } \{\sigma_2\} \text{ are both SAT}) . \quad (18)$$

The second moment entropy is then defined as $s_{2\text{nd}} \equiv \log \langle \mathcal{N}_G^2 \rangle / N$. The Chebyshev’s inequality gives then a lower bound on the satisfiability threshold via

$$\text{Prob}(\mathcal{N}_G > 0) \geq \frac{\langle \mathcal{N}_G \rangle^2}{\langle \mathcal{N}_G^2 \rangle}. \quad (19)$$

The second moment is computed in a similar manner as in [19, 32]. First we fix that in a fraction t_x of nodes the variable is occupied in both the solutions σ_1, σ_2 in (18). In a fraction t_y the variable is occupied in σ_1 and empty in σ_2 and the other way round for t_z . We sum over all possible realizations of $0 \leq t_x, t_y, t_z$ such that $\sum_{w=x,y,z} t_w \leq 1$. This is multiplied by the probability that the two configurations σ_1, σ_2 both satisfy all the constraints. After some algebra we obtain [30]:

$$s_{2\text{nd}}(t_x, t_y, t_z) = \frac{\bar{l}}{K} \log p_A(t_x, t_y, t_z) + \sum_l Q(l) \log \left\{ 1 + \sum_{w \in \{x,y,z\}} [u_w(t_x, t_y, t_z)]^l \right\}, \quad (20)$$

where $u_w(t_x, t_y, t_z)$, $w \in \{x, y, z\}$, are obtained by inverting the three equations:

$$t_w = \frac{1}{\bar{l}} \sum_l l Q(l) \frac{u_w^l}{1 + u_x^l + u_y^l + u_z^l}, \quad w = x, y, z, \quad (21)$$

and the function $p_A(t_x, t_y, t_z)$ is defined as

$$p_A(t_x, t_y, t_z) = \sum_{r_1, r_2=0}^K \delta_{A_{r_1} A_{r_2}, 1} \sum_{s=\max(0, r_1+r_2-K)}^{\min(r_1, r_2)} \binom{K}{(r_1-s)(r_2-s)s} \left(\frac{t_x}{u_x(t_x, t_y, t_z)} \right)^s \left(\frac{t_y}{u_y(t_x, t_y, t_z)} \right)^{(r_1-s)} \left(\frac{t_z}{u_z(t_x, t_y, t_z)} \right)^{(r_2-s)} (1 - t_x - t_y - t_z)^{(K-r_1-r_2+s)}. \quad (22)$$

The second moment entropy is the global maximum: $s_{2\text{nd}} = \max_{t_x, t_y, t_z} s_{2\text{nd}}(t_x, t_y, t_z)$.

For the regular graphs $Q(l) = \delta_{l,L}$ the expressions for both the first and second moment simplify considerably. For the first moment, the inverse of (17) is explicit $u = [t/(1-t)]^{1/L}$ and thus

$$s_{\text{ann reg}}(t) = \frac{L}{K} \log \left\{ \sum_{r=0}^K \delta_{A_r, 1} \binom{K}{r} [t^r (1-t)^{K-r}]^{\frac{L-1}{L}} \right\}. \quad (23)$$

For the second moment the function (21) is also explicitly reversible and the second moment entropy for regular graphs reads

$$s_{2\text{nd, reg}}(t_x, t_y, t_z) = \frac{L}{K} \log \left\{ \sum_{r_1, r_2=0}^K \sum_{s=\max(0, r_1+r_2-K)}^{\min(r_1, r_2)} \frac{K! \delta_{A_{r_1}, 1} \delta_{A_{r_2}, 1}}{(r_1-s)! (r_2-s)! s! (K-r_1-r_2+s)!} \left[t_x^s t_y^{(r_1-s)} t_z^{(r_2-s)} (1 - \sum_w t_w)^{(K-r_1-r_2+s)} \right]^{\frac{L-1}{L}} \right\}. \quad (24)$$

IV. THE PHASE DIAGRAM

A. Non-locked occupation problems

The phase diagram of the non-locked occupation problems that we have explored is qualitatively similar to the one of K-satisfiability and graph coloring studied recently in detail in [6, 8, 9, 12, 13]. We thus only briefly summarize the main findings in order to be able to appreciate the difference between the locked and the non-locked problems.

As one adds constraints to a typical non-locked problem the space of solutions undergoes several phase transitions. When the density of constraints is very small the replica symmetric solution is correct and most of the solutions lie in one cluster. As the density of constraints is increased, the point-to-set correlations, defined via the reconstruction on trees, no longer decay to zero. This is the clustering transition, at this point the space of solutions splits into exponentially many well separated (energetically or entropically) clusters. But as long as an exponential number of such clusters is needed to cover almost all the solution the observables like entropy, magnetizations, two point correlations, etc. behave as if the replica symmetric solution was still correct. This phase is called the *dynamical* 1RSB. When the constraint density is further increased the space of solutions undergoes the so-called condensation transition. In the condensed phase only a finite number of clusters is needed to cover an arbitrarily large fraction of solutions. Increasing again the density of constraints, one crosses the satisfiability transition where all the solutions disappear.

We remind at this point that in the non-locked occupation problems, where the sizes of clusters fluctuates, the survey propagation equations are not equivalent to the reconstruction on trees. More technically said the 1RSB solutions at $m = 0$ and at $m = 1$ are different, for example a non-trivial solution appears at different connectivities.

A second class of important phase transitions in the space of solutions of the non-locked problems concerns the so-called frozen variables, which might be responsible for the onset of algorithmic hardness [9]. A variable is frozen in a cluster if in all the solutions belonging to that cluster it takes the same value. A cluster is frozen if a finite fraction of variables are frozen in that cluster. A solution is frozen if it belongs to a frozen cluster. As the number of constraints is increased the clusters tend to freeze. We define two transition points. The first one, called the *rigidity* transition [9], is defined as the point where almost all solutions become frozen. The second one, the *freezing* transition, is defined as the point where strictly all solutions become frozen.

In the cavity method every cluster is associated with a solution of the BP equations. A frozen variable i is described by a marginal probability (6) which is either equal to $(\chi_0^i, \chi_1^i) = (1, 0)$ or to $(\chi_0^i, \chi_1^i) = (0, 1)$. The rigidity transition is then computed as the connectivity at which such “frozen beliefs” χ appear in the dominating clusters. If this

transition happens before the condensation transition then it is given by the onset of a nontrivial solution to the naive reconstruction, eq. (11). The rigidity transition was computed for the graph coloring in [9, 31], in the bicoloring of hyper-graphs [21], or the K-SAT in [13, 31]. The freezing transition was studied with probabilistic methods in K-SAT with large K in [33] and numerically in 3-SAT in [34].

B. Locked occupation problems

Point-like clusters — The main property which makes the locked problems special is that every cluster consists of a single configuration and has thus zero internal entropy. One way to show this is realizing that in the locked problems if $\{s_i\}$ is a satisfying configuration then

$$\psi_{s_i}^{a \rightarrow i} = 1, \quad \psi_{\neg s_i}^{a \rightarrow i} = 0, \quad (25a)$$

$$\chi_{s_i}^{i \rightarrow a} = 1, \quad \chi_{\neg s_i}^{i \rightarrow a} = 0 \quad (25b)$$

is a fixed point of BP eqs. (5a-5b). The corresponding entropy is then zero, as $Z^i = Z^a = 1$ for all i, a . In the derivation of [16] the fixed points of the belief propagation equations correspond to clusters. Thus in the locked problems every solution may be thought of as a cluster. Such a situation was previously encountered in a few problems [35, 36, 37] and called the *frozen* 1RSB because all the variables, clusters and solutions are frozen in such a case.

The clustering transition — In terms of the reconstruction on trees the situation in the locked problems is trivial because the boundary conditions on leaves always imply uniquely all the variables in the body of the tree and also the root. However one may ask what happens if the assignment of a small fraction of the variable on leaves is also forgotten – we call this the *small noise reconstruction* on trees [57]. In the non-locked problems nothing changes. In the locked problems the small noise reconstruction is not equivalent to the reconstruction. At sufficiently small connectivities the small noise reconstruction is not possible, that is if we introduce a small noise in the leaves all the information about the root is lost. In the same spirit: we showed that every solution corresponds to a fixed point of the belief propagation of the type (25), but we did not ask if such a fixed point is stable under small perturbations. If an infinitesimal fraction of messages in (25) is changed, will the iterations (5a-5b) converge back to the unperturbed fixed point or not? Again for sufficiently small connectivity it will not. This leads us back to a definition of the clustering transition which needs to be refined for the locked problems.

We thus define the *clustering transition* as the threshold for the small noise reconstruction. As all the clusters are frozen the reconstruction problem is equivalent to the naive reconstruction which deals only with the frozen variables. So for example on the ensemble of random regular graphs it is sufficient to investigate the stability of the solution $\mu_0 = \mu_1 = 1$ of eqs. (11a-11b) under iteration. It is immediate to see that if $L \geq 3$ then the solution $\mu_1 = \mu_0 = 1$ of (11a-11b) is always iteratively stable. When $L = 2$ we observed empirically that the solution $\mu_1 = \mu_0 = 1$ is not stable and the only other solutions is $\mu_1 = \mu_0 = 0$. Thus in the regular graphs ensemble of the locked problems the clustering transition is at $L = 3$.

For a general graph ensemble it is simpler to realize that as the internal entropy of clusters is zero the 1RSB solution does not depend on the value of the Parisi parameter m . Thus in particular the small noise reconstruction is equivalent to the iterative stability of the BP-like fixed point of the survey propagation equations.

We have found that, in the locked occupation problems, the SP equations (13-14), when initialized randomly, have two possible iterative fixed points:

- The trivial one: $q_*^{a \rightarrow i} = p_*^{i \rightarrow a} = 1, q_1^{a \rightarrow i} = p_1^{i \rightarrow a} = q_0^{a \rightarrow i} = p_0^{i \rightarrow a} = 0$ for all edges (ai) .
- The BP-like one: $q_*^{a \rightarrow i} = p_*^{i \rightarrow a} = 0, q^{a \rightarrow i} = \psi^{a \rightarrow i}, p^{i \rightarrow a} = \chi^{i \rightarrow a}$ for all edges (ai) , where ψ and χ is the solution of the BP equations (5a-5b) found with high probability by iterating the equations from a uniformly random initial condition.

The small noise reconstruction is then investigated, using the population dynamics, from the stability of the BP-like fixed point under iteration. If it is stable then the small noise reconstruction is possible and the phase is clustered. If it is not stable then we are in the liquid phase. From a geometric point of view, we conjecture that in the liquid phase the Hamming distance separation between solutions grows only proportionally to $\log N$; on the contrary, when small noise reconstruction is possible we expect this Hamming distance to be extensive (proportional to N).

The satisfiability threshold — The BP eqs. (5a-5b) have many fixed points. However, when we solve them iteratively starting from a random initial condition we always find the same fixed point which does not correspond to a satisfying assignment (25). We call this fixed point and its corresponding entropy the replica symmetric solution. It should actually be thought of as a fixed point of the survey propagation equations as explained in the previous paragraph. The important fact is that it gives the correct entropy (7), and also the correct marginal probabilities.

The satisfiable threshold in the locked problems is then computed as the average connectivity l_s at which the replica symmetric entropy (7) decreases to zero [35], $s(l_s) = 0$. This is the first of many quantities in the locked problems which can be computed with much smaller effort than in the non-locked problems. The condensed phase, where the space of solutions is dominated by a finite number of clusters does not exist in the locked problems, and the condensation transition coincides with the satisfiability threshold.

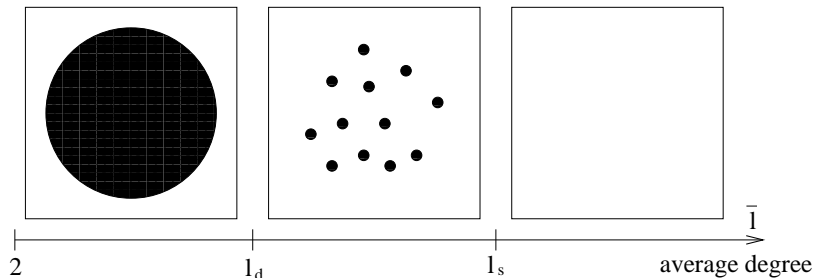


FIG. 3: Sketch of the phase diagram in the locked problems. At low constraint density $\bar{l} < l_d$ the solutions are separated by logarithmical distance but if any sort of noise is introduced this separation disappears. In the clustered phase $l_d \leq \bar{l} < l_s$ the space of solutions is made of well separated single solutions. And eventually the satisfiability transition l_s comes beyond which solutions do not exist.

Summary of the phase diagram — In contrast to the zoo of phase transitions in non-locked problems, in the locked problems we find only three phases, sketched in Fig. 3, the critical connectivity values are given in Table I.

- The liquid phase, for connectivities $\bar{l} < l_d$: In this phase the small noise reconstruction is not possible. Equivalently the BP-like iterative fixed point of the survey propagation equations is not stable. If one considers the problem at a very small temperature, the 1RSB equations have only the trivial solution, such a situation was observed previously in the perfect matching problem [18, 38]. We expect that the Hamming distance separation between solutions in this phase is only logarithmic.
- The clustered phase, for $l_d < \bar{l} < l_s$: In this phase the small noise reconstruction is possible. The BP-like iterative fixed point of the survey propagation equations is stable. The 1RSB equations have a non-trivial solution even at an infinitesimal temperature. We expect that the solutions are separated by an extensive Hamming distance, in other words there is a gap in the weight enumerator function, just like in the XOR-SAT [39]. This property is crucial in low density parity check codes [40].
- The unsatisfiable phase, for $\bar{l} > l_s$: no more solutions exist.

All the other phase transitions we described for the non-locked problems have become very simple: The clustering transition coincides with the rigidity and freezing. And the satisfiability transition coincides with the condensation one.

Finally we would like to mention the stability of the frozen 1RSB solution towards more levels of replica symmetry breaking. In more geometrical terms, one should check whether the solutions do not tend to aggregate into clusters. This is called stability of type I in the literature [41, 42, 43]. In the locked problems it is equivalent to the finiteness of the spin glass susceptibility. In all the locked occupation problems we have studied, including all those in Tab. I, we have seen that the frozen 1RSB solution is always stable in the satisfiable phase and sometimes becomes unstable at a point in the unsatisfiable phase. This means that our description of the satisfiable phase, and the determination of the thresholds l_d and l_s , should be exact.

C. The balanced LOPs

We have seen that the phase diagram in the locked problems is much simpler than in the more studied constraint satisfaction problems as the K-SAT or coloring. In this section we describe a subclass of the locked problems — the so-called *balanced* locked problems — where the situation is even simpler. In particular, the clustering and the satisfiability threshold can be determined easily, and the second moment method can be used to prove rigorously the validity of this determination of the satisfiability threshold. This makes the balanced locked problems very interesting from the mathematical point of view.

The *balanced* occupation problems are defined via the property that two random solutions are almost surely at Hamming distance $N/2 + o(N)$. This property may of course depend on the connectivity distribution $Q(l)$. A

A	name	L_s	c_d	c_s	l_d	l_s
0100	1-in-3 SAT	3	0.685(3)	0.946(4)	2.256(3)	2.368(4)
01000	1-in-4 SAT	3	1.108(3)	1.541(4)	2.442(3)	2.657(4)
00100*	2-in-4 SAT	3	1.256	1.853	2.513	2.827
01010*	4-odd-PC	5	1.904	3.594	2.856	4
010000	1-in-5 SAT	3	1.419(3)	1.982(6)	2.594(3)	2.901(6)
001000	2-in-5 SAT	4	1.604(3)	2.439(6)	2.690(3)	3.180(6)
010100	1-or-3-in-5 SAT	5	2.261(3)	4.482(6)	3.068(3)	4.724(6)
010010	1-or-4-in-5 SAT	4	1.035(3)	2.399(6)	2.408(3)	3.155(6)
0100000	1-in-6 SAT	3	1.666(3)	2.332(4)	2.723(3)	3.113(4)
0101000	1-or-3-in-6 SAT	6	2.519(3)	5.123(6)	3.232(3)	5.285(6)
0100100	1-or-4-in-6 SAT	4	1.646(3)	3.366(6)	2.712(3)	3.827(6)
0100010	1-or-5-in-6 SAT	4	1.594(3)	2.404(6)	2.685(3)	3.158(6)
0010000	2-in-6 SAT	4	1.868(3)	2.885(4)	2.835(3)	3.479(4)
0010100*	2-or-4-in-6 SAT	6	2.561	5.349	3.260	5.489
0001000*	3-in-6 SAT	4	1.904	3.023	2.856	3.576
0101010*	6-odd-PC	7	2.660	5.903	3.325	6

TABLE I: The locked cases of the occupation CSPs for $K \leq 6$. In the regular graphs ensemble the space of solutions is clustered for $L \geq L_d = 3$, and the problem is unsatisfiable for $L \geq L_s$. The values c_d and c_s are the critical parameters of the truncated Poissonian ensemble (1), the corresponding average connectivities l_d and l_s are given via eq. (2). In all these problems the replica symmetric solution is stable at least up to the satisfiability threshold. The balanced cases are marked as *, their dynamical threshold follows from (29), and their satisfiability threshold can be computed from the second moment method.

necessary condition for the problem to be balanced it that the vector A be palindromic, meaning that $A_r = A_{K-r}$. But not all the palindromic problems are balanced, the simplest such example is the 1-or-4-in-5 SAT, $A = 010010$, where the symmetry is spontaneously broken in the same way as in a ferromagnetic Ising model.

As we argued in the previous section in the locked problems the replica symmetric approach (BP) gives the exact marginal probabilities and total entropy. Therefore a problem is balanced if and only if the iterative fixed point of the BP equations (5a-5b) is such that all the beliefs are equal to $1/2$.

We do not know of any simpler general rule to decide if a problem is balanced. For $K \leq 12$, there is no exception to the following empirical rule: all the problems which can be obtained from a Fibonacci-like recursion

$$0A_K0 = A_{K+2} \quad 01A_K10 = A_{K+4} \quad (26)$$

from $A_2 = 010$ or $A_4 = 01010$ are balanced in their satisfiable phase. There are, however, other balanced locked problems which cannot be obtained this way, the simplest example is $A = 0001001000$.

Clustering threshold in the balanced LOPs: The clustering threshold is given by the small noise reconstruction, i.e. by the stability of the naive reconstruction procedure as explained in Sec. IV B. In balanced LOPs, the messages are symmetric, $\psi_0 = \psi_1 = 1/2$, and thus also the probability for the root variable to be uniquely determined by the leaves is independent of the value which has been broadcast: $\mu_0 = \mu_1 = \mu$. For a graph ensemble with excess degree distribution $q(l)$, one can write explicitly the self-consistency condition on μ :

$$\mu = \frac{2}{g_A} \sum_{r=0}^k \delta_{A_{r+1},1} \delta_{A_r,0} \binom{k}{r} \sum_{s=0}^{s_1} \binom{r}{s} \left[1 - \sum_{l=0}^{\infty} q(l)(1-\mu)^l \right]^{k-s} \left[\sum_{l=0}^{\infty} q(l)(1-\mu)^l \right]^s, \quad (27)$$

where $k = K - 1$, and $g_A = \sum_{r=0}^k \delta_{A_{r+1},1} \binom{k}{r} + \sum_{r=0}^k \delta_{A_r,1} \binom{k}{r}$. For the ensemble of graphs with truncated Poissonian degree distribution of coefficient c we derive from (3)

$$\mu = \frac{2}{g_A} \sum_{r=0}^k \delta_{A_{r+1},1} \delta_{A_r,0} \binom{k}{r} \sum_{s=0}^{s_1} \binom{r}{s} \left(\frac{1 - e^{-c\mu}}{1 - e^{-c}} \right)^{k-s} \left(\frac{e^{-c\mu} - e^{-c}}{1 - e^{-c}} \right)^s. \quad (28)$$

The clustering threshold is defined as the value of c where the fixed point $\mu = 1$ becomes unstable. One gets:

$$\frac{e^{c_d} - 1}{c_d} = K - 1 - \frac{\sum_{r=0}^{K-2} \delta_{A_{r+1,1}} \delta_{A_{r-1,0}} \delta_{A_r,0} \binom{K-1}{r}}{\sum_{r=0}^{K-2} \delta_{A_{r+1,1}} \binom{K-1}{r}}. \quad (29)$$

These values are summarized in Table I, where the balanced locked problems are marked by a *.

Satisfiability threshold in the balanced LOPs: For the balanced locked problems the replica symmetric entropy is given by:

$$s_{\text{sym}}(\bar{l}) = \log 2 + \frac{\bar{l}}{K} \log \left[2^{-K} \sum_{r=0}^K \delta_{A_r,1} \binom{K}{r} \right], \quad (30)$$

where \bar{l} is the average degree of variables. Notice the simple form of this entropy: in the balanced locked problems each added constraint destroys a fraction of solutions exactly equal to the fraction of configurations that satisfy a single constraint. The satisfiability threshold is then given by the point l_s where this entropy is zero.

Second moment method in the balanced LOPs: In all the balanced LOPs that we have considered we found numerically that the second moment entropy, (20), is exactly twice the annealed entropy (16), $2s_{\text{ann}} = s_{2\text{nd}}$. A hint that this may happen comes from the following observations:

- The annealed entropy (16) has a stationary point at $t = 1/2$ ($u = 1, x = 1$). At this stationary the entropy evaluates to (30).
- The second moment entropy (20) has a stationary point at $t_x = t_y = t_z = 1/4$ ($u_x = u_y = u_z = 1, x = y = z = 1$). At this stationary point the second moment entropy evaluates to twice the annealed entropy (30). This can be seen using the Vandermonde's combinatorial identity

$$\binom{K}{r_2} = \sum_{s=0}^{r_1} \binom{r_1}{s} \binom{K-r_1}{r_2-s}. \quad (31)$$

We checked numerically that in the balanced LOPs the global maxima of s_{ann} and $s_{2\text{nd}}$ is always given by these stationary points (the second moment entropy has another stationary point at $t_x = 1/2, t_y = t_z = 0$ or $t_x = 0, t_y = t_z = 1/2$, but at this point it is equal to the first moment entropy at $t = 1/2$). On the contrary, in the non-locked or non-balanced problems we always found another competing maximum.

If one accepted the result $2s_{\text{ann}} = s_{2\text{nd}}$, and made the reasonable assumption that the satisfiability threshold is sharp, then Chebyshev's inequality (19) would prove the correctness of the satisfiability threshold computed from (30). Therefore the full class of balanced LOPs is a candidate for a rigorous mathematical determination of the satisfiability threshold. This would be quite interesting, as it would noticeably enlarge the list of problems where the threshold is known rigorously (so far only a handful of sparse NP-complete CSPs are in this category : the 1-in- K SAT [17, 44], the $2+p$ -SAT [45, 46] and the (3,4)-UE-CSP [25]).

Let us summarize qualitatively what are the main features of the balanced locked occupation problems that make the fluctuations of the number of solutions so small that the second moment method presumably gives the exact satisfiability threshold.

- **Balancing** — It is well known that the second moment method works better if most of the weight is on the most numerous configurations (that is the half-filling ones). In the K-SAT problem several reweighting schemes were introduced in order to improve the second moment lower bounds [19, 32]. This is also the reason why the second moment bound is much sharper in the balanced NAE-SAT (bicoloring) than in the K-SAT [19].
- **Reducing fluctuations in the connectivity** — Naturally, reducing the fluctuations of the variables connectivity reduces the fluctuation of the number of solutions. Our work shows that the necessary step is not to have leaves. Fluctuating higher degrees do not really pose a problem.
- **Locked nature of the problem** — And finally the key point is the locked structure of the problem. It was remarked in [34] that the clusters-related quantities fluctuate much less than the solutions-related ones. Thus the fact that clusters do not have a fluctuating size, but size 1 is the crucial property needed to make the second moment method sharp. This is exactly what happens in the XOR-SAT problem, where the second moment becomes exact when it is restricted to the 'core' of the graph [24, 45]

V. NUMERICAL STUDIES

We shall show in this section that the LOPs, in their whole clustered phase, seem to be very hard from the algorithmic point of view. We shall illustrate this by testing and analyzing the performance of some of the best algorithms developed for random 3-satisfiability, the canonical hard constraint satisfaction problem.

Our first study uses a complete algorithm and shows that, like in other problems such as satisfiability and coloring, the hardest instances are found in the neighborhood of the satisfiability transition. We then turn to incomplete algorithms, which are aimed at finding a SAT configuration when it exists.

The best performance for incomplete algorithms is nowadays attributed to the survey propagation inspired decimation [8, 47] and the survey propagation inspired reinforcement [48]. In the random 3-SAT problem both of these algorithms were reported to work in linear time (or at most log-linear time) up to a constraint density about $\alpha = 4.252$, to be compared with the satisfiability threshold $\alpha_s = 4.267$ and the clustering transition $\alpha_d = 3.86$.

As we saw in Sec. IV B, in the LOPs, the survey propagation algorithm has no advantage over the belief propagation algorithm. We thus study the performance of the BP inspired decimation and reinforcement. The conclusions are as follows: the BP decimation fails in the LOPs even at very low connectivities; the BP reinforcement works in linear time in the non-clustered phase but fails in the clustered phase.

A. Exhaustive search results

One way to solve a LOP is to transform it into a conjunctive normal form (CNF), and use some of the open source complete solvers of the satisfiability problem. We have done such a study for the 1-or-3-in-5 SAT problem. We have generated random instances of this problem from a truncated Poisson ensemble, with M constraints. Each instance has been transformed into a satisfiability formula by mapping every constraint into $\sum_{r=0}^K \delta_{A_r,0} \binom{K}{r}$ CNF clauses: for every constraint of K variables, one creates as many CNF clauses, out of the 2^K possible clauses, as there are forbidden configurations. We have applied a branch-and-bound based open-source SAT solver called `MiniSat 1.14` [49] to test the satisfiability and to compute the running time needed by this algorithm to decide the satisfiability.

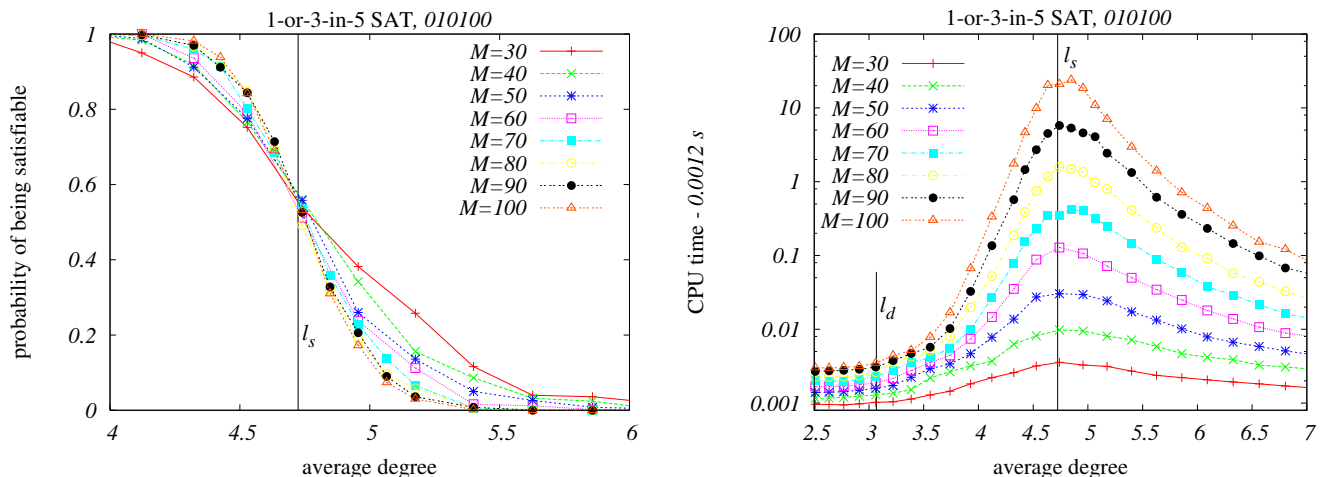


FIG. 4: *Left*: Probability that a random instance of the 1-or-2-in-5 SAT is satisfiable, versus the average degree. The probability is computed from 500 instances generated from the truncated Poisson ensemble. The vertical line shows the analytical prediction of the value of the satisfiability transition. *Right*: Median over the same 500 instances of the CPU running time of the complete algorithm `MiniSat 1.14` (we have subtracted 0.0012 seconds from the CPU time, as this is approximately where it extrapolates for small average degree and zero system size). Alternatively one could plot the number of backtracking steps, which has a qualitatively identical behavior.

Figure 4, left, shows the probability that an instance is satisfiable, plotted versus the average degree. It displays the typical behavior of a phase transition rounded by finite size effects. Figure 4, right, shows the median value of the CPU time which was used to solve an instance of the decision problem on a 2GHz MacBook laptop (note the logarithmic scale), plotted versus the average degree. The hardest instances appear around the satisfiability threshold l_s , and the time needed by the algorithm in this region clearly grows exponentially with the size. Hard satisfiable instances start to appear around $\bar{l} \simeq l_d$, although it is difficult to assert from this data where the exponential behavior

really starts. For larger system sizes it seems that the exponential behaviour starts way below the dynamical threshold l_d .

The data shows the same qualitative behavior as has been found in similar studies of satisfiability, with the difference that the relative width, $(l_s - l_d)/l_s$, of the clustered phase is larger in this case than it is in the $K = 3$ or $K = 4$ satisfiability problems. The existence of LOPs with such a broad clustered phase is an appealing feature for numerical studies. In the following sections of this paper we argue that in the locked problems the easy-hard algorithmic threshold for the best-known incomplete solvers coincides with the clustering transition l_d .

B. Decimation fails in LOPs

In BP inspired decimation one uses the knowledge of the marginal probabilities estimated from BP in order to identify the most biased variable, fix it to its most probable value, and reduce the problem. Such an algorithm usually works well even in the clustered region (for performance in K-SAT and coloring see [9, 12]). In the locked occupation problems the BP decimation fails badly. For example in the 1-or-3-in-5 SAT problem, on the truncated Poisson graphs with $M = 2 \cdot 10^4$ constraints, the probability of success is about 25% at $\bar{l} = 2$, and less than 5% at $\bar{l} = 2.3$, way below the clustering threshold $l_d \simeq 3.07$. Interestingly, the precursors of the failure of the BP decimation algorithm observed for instance in graph coloring are not present in the locked problems. In particular the BP equations converge during all the process and the normalizations in the BP equations (5a-5b) stay finite.

Although we do not know how to analyze directly the BP decimation process, the mechanisms explaining the failure of the decimation strategy can be understood using the approach of [50]. The idea is to analyze an idealized decimation process, where the variable to be fixed is chosen uniformly at random and its value is chosen according to its exact marginal probability. If its value is chosen according to the BP marginal we speak about the uniform BP decimation. If BP would give a fair approximation to the exact marginal throughout the decimation process, the uniform BP decimation should be equivalent to the ideal decimation. In the ideal decimation, the reduced problem obtained after θN steps is statistically equivalent to the reduced problem created by choosing a solution uniformly at random and revealing a fraction θ of its variables, which we now analyze, following the lines of [50].

Given an instance of the CSP, consider a solution taken uniformly at random and reveal the value of each variable with probability θ . Denote Φ the fraction of variables which either have been revealed or are directly implied by the revealed ones. We can compute $\Phi(\theta)$ using the replica symmetric cavity method (which is correct in the satisfiable phase of locked problems) as follows.

Denote by $\phi_s^{i \rightarrow b}$ the probability that a variable i is implied conditioned on the value s of the variable i and on the absence of the edge (ib) ; denote by $q_s^{a \rightarrow i}$ the probability that constraint a implies variable i to be s conditioned on: (1) variable i takes the value s in the solution we chose, (2) variable i was not revealed directly and (3) the edge (ai) is absent. Then $\phi_s^{i \rightarrow b}$ is given by:

$$\phi_s^{i \rightarrow b} = \theta + (1 - \theta) \left[1 - \prod_{a \in \partial i - b} (1 - q_s^{a \rightarrow i}) \right], \quad (32)$$

meaning that the variable i was either revealed or not, and if not it is implied if at least one of the incoming constraints implies it. We shall write the expression for $q_s^{a \rightarrow i}$ only for occupation problems on random regular graphs where the replica symmetric equation is factorized. Then $q_s^{a \rightarrow i}$ and $\phi_s^{i \rightarrow b}$ are independent of a, b, i : $q_s^{a \rightarrow i} = q_s$ and $\phi_s^{i \rightarrow b} = \phi_s$. The conditional probability q_s is the ratio of the probability that variable i takes the value s and is implied by the constraint a on one hand, and the probability that variable i takes the value s on the other hand:

$$q_1 = \frac{1}{\psi_1 Z^{\text{reg}}} \sum_{r=0}^k \delta_{A_r, 0} \delta_{A_{r+1}, 1} \binom{k}{r} (\psi_1)^{lr} (\psi_0)^{l(k-r)} \sum_{s=0}^{s_1} \binom{r}{s} \phi_0^{k-r} \phi_1^{r-s} (1 - \phi_1)^s, \quad (33a)$$

$$q_0 = \frac{1}{\psi_0 Z^{\text{reg}}} \sum_{r=0}^k \delta_{A_r, 1} \delta_{A_{r+1}, 0} \binom{k}{r} (\psi_1)^{lr} (\psi_0)^{l(k-r)} \sum_{s=0}^{s_0} \binom{k-r}{s} \phi_1^r \phi_0^{k-r-s} (1 - \phi_0)^s, \quad (33b)$$

where $l = L - 1$, $k = K - 1$. The sum over r goes over all the possible numbers of 1's being assigned on the incoming variables, and the numbers ψ_0, ψ_1 are the cavity probabilities, solutions of the BP equations (9a-9b). The indices s_1, s_0 in the second sum of both equations are the largest possible but such that $s_1 \leq r$, $s_0 \leq K - 1 - r$, and $\sum_{s=0}^{s_1} A_{r-s} = 0$, $\sum_{s=0}^{s_0} A_{r+1+s} = 0$. The terms $\phi_0^{k-r} \phi_1^{r-s} (1 - \phi_1)^s$ and $\phi_1^r \phi_0^{k-r-s} (1 - \phi_0)^s$ are the probabilities that a sufficient number of incoming variables was revealed such that the out-coming variable is implied (not conditioned on its value). Z^{reg} is the normalization in (9a-9b).

Iterations of eqs. (32-33) with the initial condition $\phi = \theta$ give us the fixed point for q_0, q_1 . The total probability that a variable is fixed is then computed as

$$\Phi(\theta) = \theta + (1 - \theta) \{ \mu_1 [1 - (1 - q_1)^L] + \mu_0 [1 - (1 - q_0)^L] \}, \quad (34)$$

where μ_0, μ_1 are the total BP marginals, $\mu_s = \psi_s^L / (\psi_0^L + \psi_1^L)$. Notice the analogy between eqs. (33b-33a) and the equations for the naive reconstruction (11b-11a).

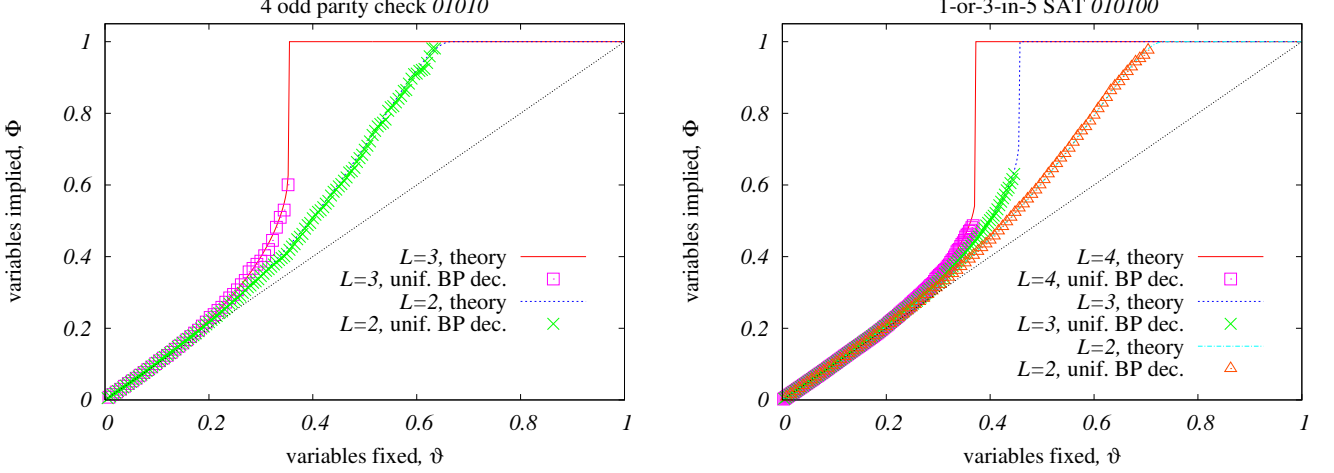


FIG. 5: Analytical and numerical study of the BP inspired uniform decimation. The number of variables which are directly implied, $\Phi(\theta)$, is plotted against the number of fixed variables θ in two of the LOPs on the regular graph ensemble with connectivity L .

In Fig. 5 we compare the function $\Phi(\theta)$ obtained from the analytical study of ideal decimation (34) with the experimental performance of the uniform BP decimation. Before the failure of the decimation algorithm (when a contradiction is encountered) the two curves are in very good agreement. This study shows two different reasons for the failure of the algorithm:

- Avalanche of direct implications – In some cases the function $\Phi(\theta)$ has a discontinuity at a certain spinodal point θ_s (e.g. $\theta_s \approx 0.46$ at $L = 3$ for the 1-or-3-in-5 SAT problem). For $\theta < \theta_s$, fixing one variable generates a finite number of direct implications. As the loops are of order $\log N$ these implications never lead to a contradiction. At the spinodal point θ_s , fixing one more variable generates an extensive avalanche of direct implications. Small (order $1/N$) errors in the previously used BP marginals may thus lead to a contradiction. This indeed happens in almost all the runs we have done. For more detailed discussion see [50].
- No more free variables – The second reason for the failure is specific to the locked problems, more precisely to the problems where $\phi_0 = \phi_1 = 1$ is a solution of eqs. (32-33). In these cases it may happen that the function $\Phi(\theta) \rightarrow 1$ at some $\theta_1 < 1$ (e.g. $\theta_1 \approx 0.73$ at $L = 4$ for the 1-or-3-in-5 SAT problem). In other words if we reveal a fraction $\theta > \theta_1$ of variables from a random solution, the reduced problem will be compatible with only that given solution. Again, if there has been a little error in the previously fixed variables, the BP uniform decimation ends up in a contradiction. If on the contrary the function $\Phi(\theta)$ reaches the value 1 only for $\theta = 1$ then the residual entropy is positive and there might be at each step some space to correct previous small errors, demonstrated on a non-locked problem in Fig. 6.

These two reasons of failure of the BP uniform decimation seem to be of quite different nature. But they have one property in common. As the point of failure is approached we observe that almost all the variables which are being fixed were already implied. The same sign of failure can be observed also in the maximal BP decimation. In Fig. 6 we compare the two procedures. On the x -axes we plot the number of variables which could have taken both the values just before they were fixed. On the y -axes the number of variables which could take only one value before they were fixed plus the number of implied variables. The failure of both the versions of the BP decimation algorithms is then related to the divergence of the derivative of the function $y(x)$.

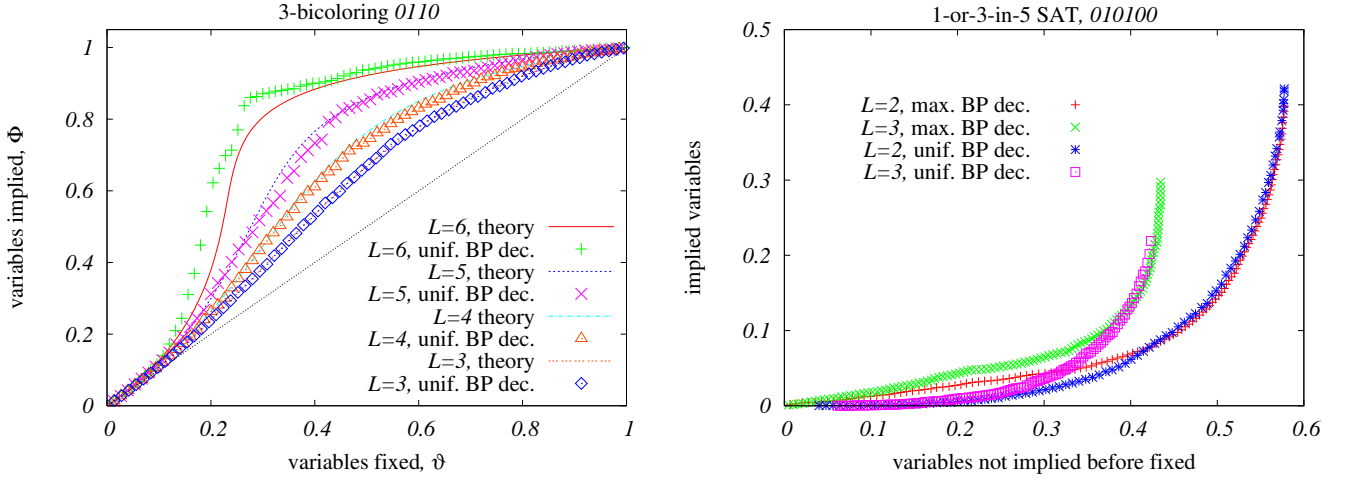


FIG. 6: Left: For comparison, the BP uniform decimation works well on the non-locked problems, the example is for bicoloring. Right: Comparison of the uniform BP decimation with the maximal BP decimation. The number of variables which are directly implied or were directly implied before being fixed is plotted against the number of variables which were free just before being fixed. The behavior of the two decimation strategies is similar. The divergence of the derivative of this function marks the point of failure.

C. The BP reinforcement algorithm

BP reinforcement is currently the most efficient way of using the BP equations in a solver. It was originally introduced in [48], and has also been used in [21, 51]. The main idea is to add an ‘external bias’ $\mu_{s_i}^i$ which biases the variable i in the direction of the marginal probability computed from the BP messages. This modifies BP eq. (5b) to

$$\psi_{s_i}^{a \rightarrow i} = \frac{1}{Z^{a \rightarrow i}} \sum_{\{s_j\}} \delta_{A_{s_i} + \sum_j s_j, 1} \prod_{j \in \partial a - i} \chi_{s_j}^{j \rightarrow a}, \quad (35a)$$

$$\chi_{s_i}^{i \rightarrow a} = \frac{1}{Z^{i \rightarrow a}} \mu_{s_i}^i \prod_{b \in \partial i - a} \psi_{s_i}^{b \rightarrow i}, \quad (35b)$$

We remind that the belief on variable i (the BP estimate of its marginal) $\chi_{s_i}^i$, without taking into account the bias μ , is given by eq. (6).

We tried several implementations of how the external bias $\mu_{s_i}^i$ is updated and found the best performance for the following one

$$\mu_1^i = \pi, \quad \mu_0^i = 1 - \pi, \quad \text{if } \chi_0^i > \chi_1^i, \quad (36a)$$

$$\mu_1^i = 1 - \pi, \quad \mu_0^i = \pi, \quad \text{if } \chi_0^i \leq \chi_1^i, \quad (36b)$$

where $0 \leq \pi \leq 1/2$ is a parameter which needs to be optimized. In the iterative update of the BP reinforcement, the external bias is not updated at every BP iteration, but only with probability

$$p(t) = 1 - (1 + t)^{-\gamma}. \quad (37)$$

where t is the time step, and γ a parameter to be optimized. The pseudocode of the algorithm is then as follows.

BP-REINFORCEMENT(T, γ, π)

- 1 Initialize $\mu_{s_i}^i$ and $\psi_{s_i}^{a \rightarrow i}$ randomly;
- 2 $t \leftarrow 0$;
- 3 Compute the current configuration $r_i = \operatorname{argmax}_{s_i} \mu_{s_i}^i$;
- 4 **repeat** Make one sweep of the BP iterations (35a-35b);
- 5 update every bias $\mu_{s_i}^i$ with probability $p(T)$ according to (36a-36b);
- 6 Update $r_i = \operatorname{argmax}_{s_i} \mu_{s_i}^i$;
- 7 $t \leftarrow t + 1$;
- 8 **until** $\{r\}$ is a solution or $t > T$;

This algorithm depends on two empirical parameters, γ and μ . We generally use $\gamma = 0.1$. The optimization of the bias strength π is crucial. Empirically we observed three different regimes:

- (a) $\pi_{\text{BP-like}} < \pi < 0.5$: When the bias is weak, BP-REINFORCEMENT converges very fast to a BP-like fixed point, the values of the local fields do not point towards any solution. On contrary many constraints are violated by the final configuration $\{r_i\}$.
- (b) $\pi_{\text{conv}} < \pi < \pi_{\text{BP-like}}$: BP-REINFORCEMENT converges to a solution $\{r_i\}$.
- (c) $0 < \pi < \pi_{\text{conv}}$: When the bias is too strong, BP-REINFORCEMENT does not converge. And many constraints are violated by the configuration $\{r_i\}$ which is reached after T_{max} steps.

When the constraint density in the CSP is large the regime (b) disappears and $\pi_{\text{conv}} = \pi_{\text{BP-like}}$. Clearly, the goal is to find $\pi_{\text{conv}} < \pi < \pi_{\text{BP-like}}$. The point $\pi_{\text{BP-like}}$ is very easy to find, because for larger π the convergence of BP-REINFORCEMENT to a BP-like fixed point takes place in just a few sweeps. Thus in all the runs we chose π to be just below $\pi_{\text{BP-like}}$. The value of π chosen in this way does not seem to depend on the size of the system, but it depends slightly on the constraint density.

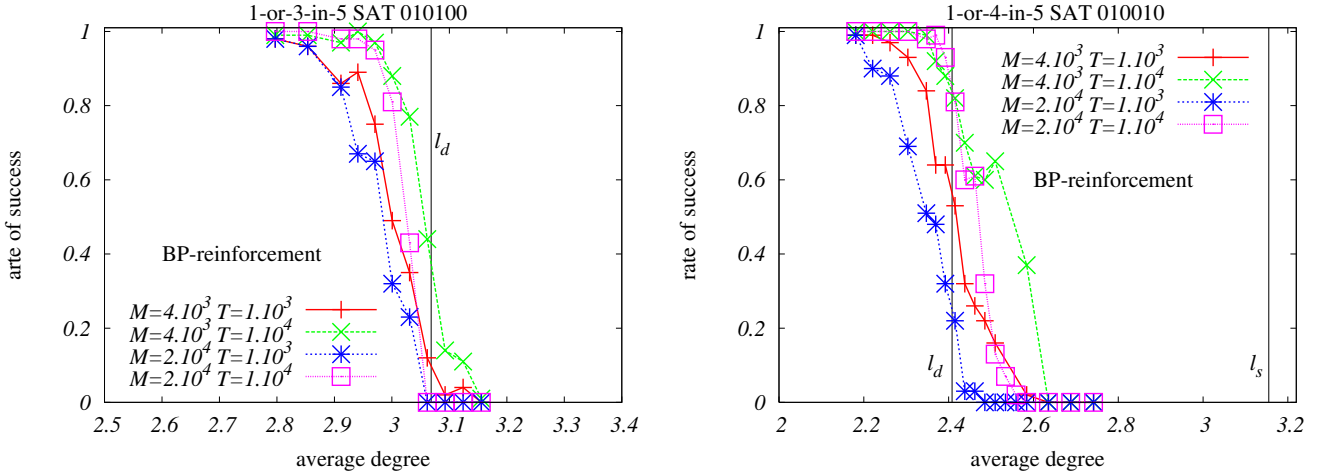


FIG. 7: Performance of the BP reinforcement on two of the locked occupation problems. Probability of success versus average connectivity. Left: $A = 010100$, the optimal parameters: $\gamma = 0.1$, $\pi = 0.28$ for $2.79 \leq \bar{l} \leq 2.95$, $\pi = 0.30$ for $2.97 \leq \bar{l} \leq 3.13$, $\pi = 0.31$ for $c = 3.15$. Right: $A = 010010$ with $\gamma = 0.1$, $\pi = 0.34$. The different curves are for two different system sizes and two different maximal running times. The algorithm performs well only up to a connectivity close to the clustering transition ($l_d = 3.07$ resp. $l_d = 2.41$) to be compared with the satisfiability threshold $l_s = 4.72$ resp. $l_s = 3.16$). Qualitatively similar result were observed for all the other locked occupation problems we studied.

We tested the BP-REINFORCEMENT algorithm on the locked occupation CSPs, the results are shown in Fig. 7. The fraction of successful runs on different system sizes and for different maximal running times is plotted as a function of the mean variable connectivity. Our data suggest that the algorithm is successful only in the liquid phase, and fails in the clustered (that is also frozen) region. Similar results can be obtained with other algorithms; for instance the performance of stochastic local search was reported in [10].

The clustered phase is thus extremely hard and instances of the locked problems can serve as benchmarks for new solvers. In fact, some of the hardest benchmarks of the K-satisfiability problem are based on a well known LOP, XOR-SAT (with some additional non-linear function nodes which rule out the Gaussian elimination solvers) [52, 53].

In the non-locked problems the very same implementation of the BP reinforcement is able to find solutions inside the clustered region, Fig. 8 left shows the performance for $A = 011010$. This is in qualitative agreement with results for the K-SAT [48], coloring [9, 12] or bicoloring problems [21].

It is not known how one can characterize from a geometrical point of view the connectivity threshold where BP reinforcement algorithms stop to be efficient in the non-locked problems. It has been found in [21] that even the rigid phase where almost all solutions are frozen may be algorithmically easy. Fig. 8 confirms this statement for the problem $A = 0110100$ on the regular ensemble with $L = 8$. The ratio of success of the BP reinforcement (with 3 restarts) is close to one, and basically independent of system size, while it can be seen from (11) that this problem is in the rigid phase. On the other hand the fraction of found solutions which are frozen (have a nontrivial whitening core [34, 54]) goes to zero as the system size is growing, in agreement with the results of [21]. Thus the question of where is the easy/hard threshold in the non-locked problems remains open.

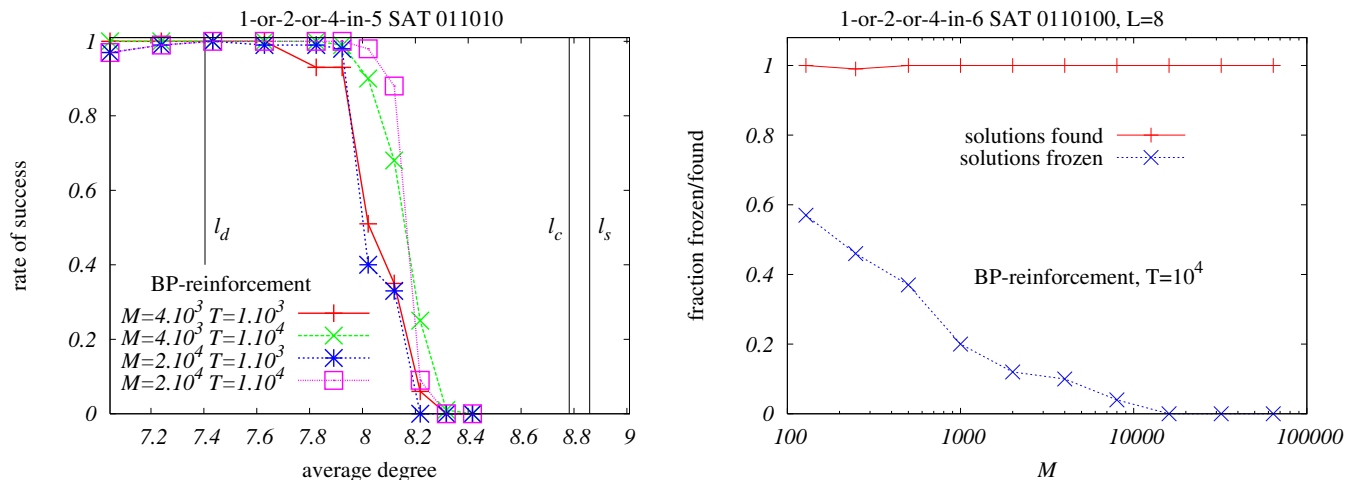


FIG. 8: Left: Performance of the BP reinforcement on one of the non-locked problems, $A = 010110$. Parameters: $\gamma = 0.1$; $\pi = 0.40$ for $7.0 \leq \bar{l} \leq 7.8$, $\pi = 0.42$ for $7.9 \leq \bar{l} \leq 8.0$, $\pi = 0.44$ for $8.1 \leq \bar{l} \leq 8.4$. The implementation of the algorithm is the same as for the locked problems in Fig. 7. Here solutions are found up to about a half of the clustered region, $l_d = 7.40$. The condensation $l_c = 8.78$ and the satisfiability $l_s = 8.86$ transition are also marked. Right: The $A = 0110100$ at regular graphs of $L = 8$ is in the rigid phase, that is almost all solutions belong to frozen clusters. Yet the BP reinforcement ($\gamma = 0.1$, $\pi = 0.36$) finds a solution almost surely (after 3 restarts) – the red curve. The blue curve gives a fraction of how many of the solutions found belonged to a frozen cluster. We see that asymptotically we never find the frozen solutions.

VI. CONCLUSION

We studied the class of occupation CSPs on which we illustrated the difference between locked and non-locked CSPs. The point-like nature of clusters in LOPs is responsible for all of these differences. Our finding may be summarized as: "Locked problems are extremely simple and extremely hard." The simplicity comes at the level of the phase diagram, which can be computed by the cavity method much more easily than in the general CSP. In certain cases some non-trivial quantities are probably amenable to a rigorous study along the lines that we sketched – as for example the satisfiability threshold in the balanced locked problems. The hardness is algorithmic, some algorithms – as the BP decimation – fail completely, and even the best known algorithms are not able to find solutions in the clustered phase of the locked problems. Their simple description and algorithmic hardness makes the locked problems challenging for developments of new algorithms as well as for better theoretical understanding on the origin of hardness.

There are several clear directions in which this work should be extended. The planted ensembles of LOPs should be studied in order to provide hard benchmarks where the existence of a solution would be guaranteed. On the mathematical side the rigorous proof of the second moment method giving the satisfiability threshold in the balanced locked problems should be worked out. One may investigate if the location of the clustering threshold can be proven rigorously using the small noise reconstruction in the lines of [55], or bounding the weight enumerator function in the lines of [39]. Also it will be interesting to study (at least numerically) the dynamics at finite temperature, as it might provide further insight into the connection between the dynamics of algorithms and structure of solutions. Finally the distance properties between solutions in locked CSPs makes them interesting candidates for the development of nonlinear error correcting codes or compression schemes.

Acknowledgment

We thank Florent Krzakala, Cris Moore, Thierry Mora and Guilhem Semerjian for very useful discussions.

-
- [1] S. A. Cook, in *Proc. 3rd STOC* (ACM, New York, NY, USA, 1971), pp. 151–158.
 - [2] M. Garey and D. Johnson, *Computers and intractability: A guide to the theory of NP-completeness* (Freeman, San Francisco, 1979).
 - [3] P. Cheeseman, B. Kanefsky, and W. M. Taylor, in *Proc. 12th IJCAI* (Morgan Kaufmann, San Mateo, CA, USA, 1991), pp. 331–337.

- [4] D. G. Mitchell, B. Selman, and H. J. Levesque, in *Proc. 10th AAAI* (AAAI Press, Menlo Park, California, 1992), pp. 459–465.
- [5] M. Mézard, G. Parisi, and M. A. Virasoro, *Spin-Glass Theory and Beyond*, vol. 9 of *Lecture Notes in Physics* (World Scientific, Singapore, 1987).
- [6] M. Mézard, G. Parisi, and R. Zecchina, *Science* **297**, 812 (2002).
- [7] G. Biroli, R. Monasson, and M. Weigt, *Eur. Phys. J. B* **14**, 551 (2000).
- [8] M. Mézard and R. Zecchina, *Phys. Rev. E* **66**, 056126 (2002).
- [9] L. Zdeborová and F. Krzakala, *Phys. Rev. E* **76**, 031131 (2007).
- [10] L. Zdeborová and M. Mézard, *Phys. Rev. Lett.* **101**, 078702 (2008).
- [11] R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina, *Phys. Rev. Lett.* **89**, 268701 (2002).
- [12] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová, *Proc. Natl. Acad. Sci. U.S.A* **104**, 10318 (2007).
- [13] A. Montanari, F. Ricci-Tersenghi, and G. Semerjian, *J. Stat. Mech.* p. P04004 (2008).
- [14] T. Mora, Ph.D. thesis, Université Paris-Sud (2007), <http://tel.archives-ouvertes.fr/tel-00175221/en/>.
- [15] F. R. Kschischang, B. Frey, and H.-A. Loeliger, *IEEE Trans. Inform. Theory* **47**, 498 (2001).
- [16] M. Mézard and A. Montanari, *Information, Physics, Computation: Probabilistic approaches* (Cambridge University Press, Cambridge, 2008), in preparation: www.lptms.u-psud.fr/membres/mezard/.
- [17] J. Raymond, A. Sportiello, and L. Zdeborová, *Phys. Rev. E* **76**, 011101 (2007).
- [18] L. Zdeborová and M. Mézard, *J. Stat. Mech.* **5**, P05003 (2006).
- [19] D. Achlioptas and C. Moore, *SIAM Journal on Computing* **36**, 740 (2006).
- [20] T. Castellani, V. Napolano, F. Ricci-Tersenghi, and R. Zecchina, *J. Phys. A* **36**, 11037 (2003).
- [21] L. Dall’Asta, A. Ramezanpour, and R. Zecchina, *Phys. Rev. E* **77**, 031118 (2008).
- [22] E. Marinari and G. Semerjian, *J. Stat. Mech.* **6**, P06019 (2006).
- [23] T. J. Schaefer, in *Proc. 10th STOC* (ACM, New York, NY, USA, 1978), pp. 216–226.
- [24] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina, *J. Stat. Phys.* **111**, 505 (2003).
- [25] H. Connamacher and M. Molloy, in *45th Symposium on Foundations of Computer Science (FOCS 2004)*, 17-19 October 2004, Rome, Italy, *Proceedings* (IEEE Computer Society, 2004), ISBN 0-7695-2228-9.
- [26] M. Mézard and G. Parisi, *Eur. Phys. J. B* **20**, 217 (2001).
- [27] J. Pearl, in *Proceedings American Association of Artificial Intelligence National Conference on AI* (Pittsburgh, PA, USA, 1982), pp. 133–136.
- [28] M. Mézard and A. Montanari, *J. Stat. Phys.* **124**, 1317 (2006).
- [29] E. Mossel, *Ann. Appl. Probab.* **11**, 285 (2001).
- [30] L. Zdeborová, Ph.D. thesis, Université Paris-Sud (2007), arXiv:0806.4112v1 [cond-mat.stat-mech].
- [31] G. Semerjian, *J. Stat. Phys.* **130**, 251 (2008).
- [32] D. Achlioptas, A. Naor, and Y. Peres, *Nature* **435**, 759 (2005).
- [33] D. Achlioptas and F. Ricci-Tersenghi, in *Proc. of 38th STOC* (ACM, New York, NY, USA, 2006), pp. 130–139.
- [34] J. Ardelius and L. Zdeborová, *Phys. Rev. E (Rapid Com.)* p. 040101(R) (2008).
- [35] W. Krauth and M. Mézard, *J. Phys. France* **50**, 3057 (1989).
- [36] O. C. Martin, M. Mézard, and O. Rivoire, *Phys. Rev. Lett.* **93**, 217205 (2004).
- [37] O. C. Martin, M. Mézard, and O. Rivoire, *J. Stat. Mech.* p. P09006 (2005).
- [38] H. Zhou and Z. Ou-Yang (2003), preprint arXiv:cond-mat/0309348v1 [cond-mat.dis-nn].
- [39] T. Mora and M. Mézard, *J. Stat. Mech.* **10**, P10007 (2006).
- [40] C. Di, T. J. Richardson, and R. Urbanke, *IEEE Trans. Inform. Theory* **52**, 4839 (2006).
- [41] A. Montanari and F. Ricci-Tersenghi, *Eur. Phys. J. B* **33**, 339 (2003).
- [42] O. Rivoire, G. Biroli, O. C. Martin, and M. Mézard, *Eur. Phys. J. B* **37**, 55 (2004).
- [43] F. Krzakala and L. Zdeborová, *Eur. Phys. Lett.* **81**, 57005 (2008).
- [44] D. Achlioptas, A. Chtcherba, G. Istrate, and C. Moore, in *SODA ’01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms* (SIAM, Philadelphia, PA, USA, 2001), pp. 721–722.
- [45] S. Cocco, O. Dubois, J. Mandler, and R. Monasson, *Phys. Rev. Lett.* **90**, 047205 (2003).
- [46] D. Achlioptas, L. M. Kirousis, E. Kranakis, and D. Krizanc, *Theoretical Computer Science* **256**, 109 (2001).
- [47] G. Parisi (2003), arXiv:cs/0301015.
- [48] J. Chavas, C. Furtlehner, M. Mézard, and R. Zecchina, *J. Stat. Mech.* p. P11016 (2005).
- [49] N. Eén and N. Sorensson, in *Theory and Applications of Satisfiability Testing* (Springer Berlin, Heidelberg, 2004), pp. 502–518.
- [50] A. Montanari, F. Ricci-Tersenghi, and G. Semerjian (2007), to appear in Proceedings of Allerton 2007. Preprint: arXiv:0709.1667v1 [cs.AI].
- [51] A. Braunstein and R. Zecchina, *Physical Review Letters* **96**, 030201 (2006).
- [52] H. Haanpää, M. Järvisalo, P. Kaski, and I. Niemelä, *Journal on Satisfiability, Boolean Modeling and Computations* **2**, 27 (2006).
- [53] W. Barthel, A. K. Hartmann, M. Leone, F. Ricci-Tersenghi, M. Weigt, and R. Zecchina, *Phys. Rev. Lett.* **88**, 188701 (2002).
- [54] A. Braunstein and R. Zecchina, *Journal of Statistical Mechanics: Theory and Experiment* p. P06007 (2004).
- [55] A. Sly (2008), arXiv:0802.3487v1 [math.PR].
- [56] S. Janson and E. Mossel, *Ann. Probab.* **32**, 2630 (2004).

[57] The related, but different, concept of *robust reconstruction* was studied in [56]. Whereas in the small noise reconstruction one investigates the effect of an infinitesimal noise-rate in the robust reconstruction one studies the effect of a noise-rate arbitrarily close to 1.